

BN-1 C 言語開発キット

BN-1 API リファレンス

Version 1.00

株式会社バンダイ

有限会社トラスト・テクノロジー

目次

第 1 章	はじめに.....	1
第 2 章	ヘッダファイル.....	4
2.1	マクロの定義.....	4
2.2	構造体.....	5
第 3 章	関数リファレンス.....	6
3.1	ユーティリティ関数.....	6
(1)	BN1WaitSec.....	6
(2)	BN1WaitMilliSec.....	6
(3)	BN1CharToHexStr.....	7
(4)	BN1SetTraceLevel.....	7
(5)	BN1TraceOut.....	8
(6)	BN1TraceIn.....	9
(7)	BN1BreakOut.....	10
3.2	グラフィックアイ関数.....	11
(1)	BN1InitEye.....	11
(2)	BN1PutEye.....	12
(3)	BN1StopEye.....	12
(4)	BN1SetEyeColor.....	13
(5)	BN1SetEyeBrightness.....	14
(6)	BN1SetEyePattern.....	15
(7)	BN1PutEyePattern.....	16
(8)	BN1PlayEyeAnimation.....	17
(9)	BN1StopEyeAnimation.....	18
(10)	BN1SetEyeSpeed.....	18
3.3	サウンド関数.....	19
(1)	BN1InitSound.....	19
(2)	BN1PlaySound.....	19
(3)	BN1StartSound.....	20
(4)	BN1StopSound.....	20
(5)	BN1SetSoundVolume.....	21
3.4	モーション関数.....	22
(1)	BN1DoMotion.....	22
(2)	BN1StopMotion.....	22
(3)	BN1DoPosition.....	23

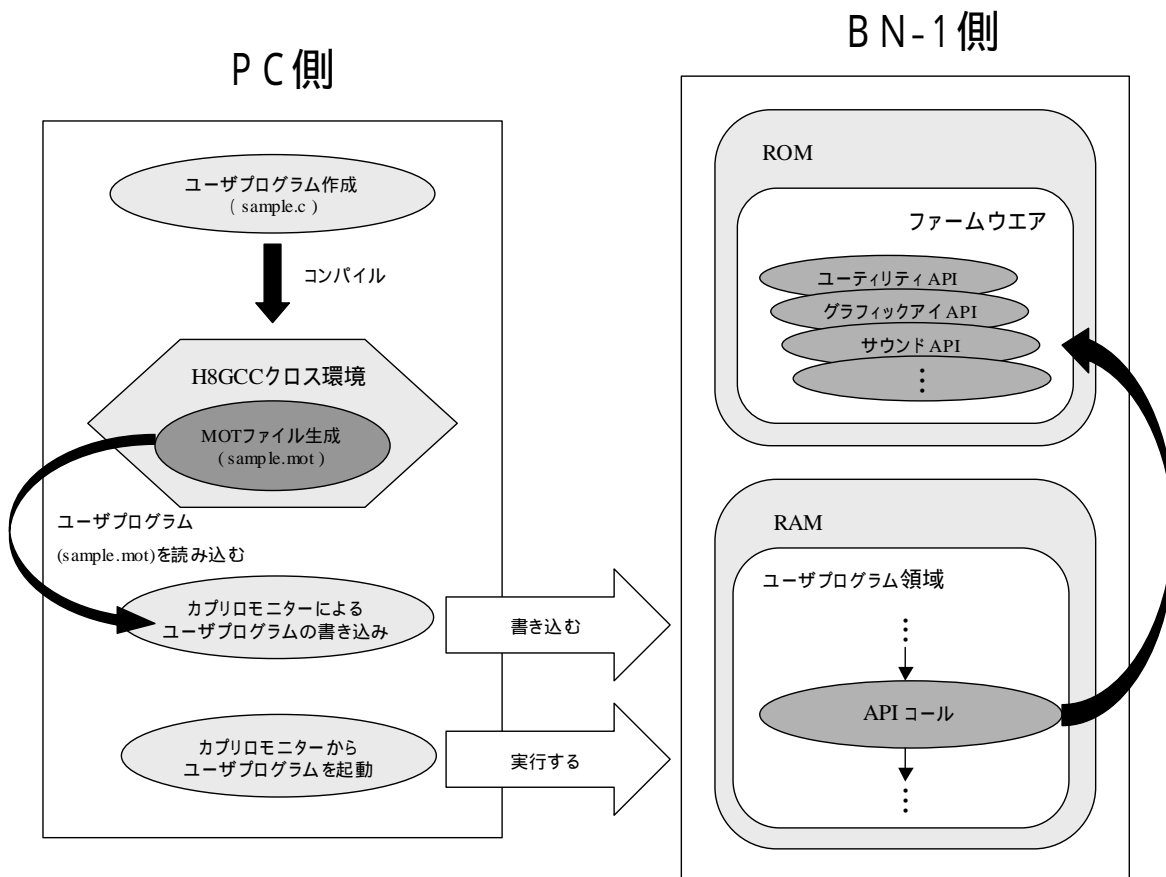
(4) BN1DriveRightArm	24
(5) BN1DriveLeftArm.....	24
(6) BN1DriveWaist	25
(7) BN1DriveRightWheel	25
(8) BN1DriveLeftWheel	26
(9) BN1StopRightArm.....	26
(10) BN1StopLeftArm	27
(11) BN1StopWaist	27
3.5 センサー関数	28
(1) BN1InitSensor	28
(2) BN1GetSensorInfo	29
(3) BN1SensStroke	30
(4) BN1SensKnock.....	30
3.6 タイマー割込み関数	31
(1) BN1SetTimer16	31
(2) BN1SetTimer8	33
3.7 通信関数	34
(1) BN1WriteCOM.....	34
(2) BN1ReadCOM.....	35
付録	36
付録A センサー解説	36
付録B データ解説.....	40
(1) モーションデータ	40
(2) サウンドデータ	43
(3) グラフィックアイ (アニメ - ション) データ	44

第 1 章 はじめに

本マニュアルでは、BN-1 API(Application Programming Interface)の仕様について解説します。

BN-1 API とは

BN-1 API は、RAM領域に配置されるユーザプログラムからコールされることにより、BN-1 本体の容易なコントロールを可能にするライブラリです。BN-1 API の制御ロジックはフラッシュROMにファームウェアとして格納されており、次項より解説する API インターフェースにより、ユーザプログラム領域を圧迫することなく利用できます。



* MOT ファイル・・・ H 8 用プログラムのバイナリイメージをモトローラ S2 ファイル形式に変換した ASCII テキストファイルです。

図 1 . ユーザプログラム実行環境

BN-1 API の機能を大別すると、以下の 7 種類に分類されます。

- 1 . ユーティリティ関数
- 2 . グラフィックアイ関数
- 3 . サウンド関数
- 4 . モーション関数
- 5 . センサー関数
- 6 . タイマー割込み関数
- 7 . 通信関数

* これらの関数を使用するには、

ユーザプログラムのソースファイルに BN-1 API ヘッダファイル `bn1api.h` をインクルードする必要があります。`/usr/local/h8/include/bn1/` 以下に配置されていますので、ソースファイル内で

```
#include <bn1api.h>
```

と宣言をし、コンパイル時のオプションで `I/usr/local/h8/include/bn1` と場所を指定してください。

また、オブジェクトファイルのリンク時にも BN-1 API ライブラリ `libbn1api.a` をリンクする必要があります。`/usr/local/h8/lib/bn1/` 以下に配置されていますので、リンク時のオプションで `-L/usr/local/h8/lib/bn1` と場所を指定し、`-lbn1api` とリンクライブラリを指定することで利用できます。

表 1 . 関数の一覧表

関数名	機能
ユーティリティ関数	
BN1WaitSec	一定時間ウエイトをかけて一時停止させます。(秒単位)
BN1WaitMilliSec	一定時間ウエイトをかけて一時停止させます。(ミリ秒単位)
BN1CharToHexStr	1バイトのchar型の数値を16進表記の文字列(2バイト)にします。
BN1SetTraceLevel	カブリロモニター(capterm)のトレースレベルを設定します。
BN1TraceOut	BN1TraceOut 関数の第1引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、カブリロモニター(capterm)にトレース用文字列を出力します。
BN1TraceIn	BN1TraceIn 関数の第1引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、カブリロモニター(capterm)から文字列を入力します。
BN1BreakOut	BN1BreakOut 関数の引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、プログラムを一時中断し、カブリロモニター(capterm)に続行を尋ねます。
グラフィックアイ関数	
BN1InitEye	グラフィックアイ機能を初期化します。
BN1PutEye	グラフィックアイを出力します。
BN1StopEye	グラフィックアイを停止(消去)します。
BN1SetEyeColor	グラフィックアイの色を設定します。
BN1SetEyeBrightness	グラフィックアイの明るさを設定します。
BN1SetEyePattern	グラフィックアイのユーザ定義パターン(外字)を設定します。
BN1PutEyePattern	BN1SetEyePattern 関数で定義したグラフィックアイのユーザ定義パターン(外字)を出力します。
BN1PlayEyeAnimation	グラフィックアイの定義済みアニメーションを実行します。
BN1StopEyeAnimation	グラフィックアイの定義済みアニメーションを停止します。
BN1SetEyeSpeed	グラフィックアイの定義済みアニメーションのスピードを設定します。
サウンド関数	
BN1InitSound	サウンド機能を初期化します。
BN1PlaySound	サウンドの指定番号を演奏します。
BN1StartSound	サウンドの演奏を開始します。
BN1StopSound	サウンドの演奏を停止します。
BN1SetSoundVolume	サウンドのボリュームを設定します。
モーション関数	
BN1DoMotion	モーションの指定番号を実行します。
BN1StopMotion	モーションを停止します。
BN1DoPosition	指定のポジションにします。
BN1DriveRightArm	右腕を単独で動かします。
BN1DriveLeftArm	左腕を単独で動かします。
BN1DriveWaist	腰を単独で動かします。
BN1DriveRightWheel	右輪を単独で動かします。
BN1DriveLeftWheel	左輪を単独で動かします。
BN1StopRightArm	右腕を停止します。(弛緩)
BN1StopLeftArm	左腕を停止します。(弛緩)
BN1StopWaist	腰を停止します。(弛緩)
センサー関数	
BN1InitSensor	センサー機能を初期化します。
BN1GetSensorInfo	センサー情報を得ます。
BN1SensStroke	撫でられた状態を感知します。
BN1SensKnock	叩かれた状態を感知します。
タイマー割込み関数	
BN1SetTimer16	タイマーによる割込み関数を設定します。 (16ビットタイマー 10us 単位で tm count で起動)
BN1SetTimer8	タイマーによる割込み関数を設定します。 (8ビットタイマー 166ms 単位で tm count で起動)
通信関数	
BN1WriteCOM	シリアル通信でデータを書き込みます。
BN1ReadCOM	シリアル通信でデータを読み込みます。

第2章 ヘッダファイル

ヘッダファイルではマクロの定義と関数のプロトタイプ宣言を行っています。
BN-1API 関数を使用するには `bn1api.h` をインクルードする必要があります。
以下に `bn1api.h` で定義されているマクロの説明をします。

2.1 マクロの定義

定義名一覧	説明
<code>ABS(val)</code>	引数の絶対値を返します
<code>NULL</code>	ヌル値を示す値。0 に設定
<code>TRUE</code>	真理値型 <code>Bool_t</code> で使用する真の値。0xff に設定
<code>FALSE</code>	真理値型 <code>Bool_t</code> で使用する偽の値。0x00 に設定
<code>BN1_PORT_BREAST</code>	BN-1 の胸部ハッチにあるシリアルポートの番号
<code>Bool_t</code>	<code>typedef</code> で定義した <code>unsigned char</code> 型

2.2 構造体

`bn1api.h` では、センサー情報を格納する `SensorInfo_t` (typedef をした構造体) を宣言しています。 `SensorInfo_t` のメンバは以下の通りです。

センサー情報一覧

(1)	<code>unsigned char</code>	<code>slant6</code>	傾きセンサー 6 方位
(2)	<code>Bool_t</code>	<code>overturnwarn</code>	転倒警告センサー
(3)	<code>unsigned char</code>	<code>infrared</code>	赤外線センサー
(4)	<code>unsigned char</code>	<code>nikukyu</code>	肉球センサー
(5)	<code>Bool_t</code>	<code>shake</code>	シェイクセンサー
(6)	<code>unsigned char</code>	<code>chira1</code>	チラチラセンサー(前後左右)
(7)	<code>unsigned char</code>	<code>chira2</code>	チラチラセンサー(のど)
(8)	<code>unsigned char</code>	<code>mic</code>	マイクセンサー
(9)	<code>int</code>	<code>leftarm</code>	左腕位置センサー
(1 0)	<code>int</code>	<code>rightarm</code>	右腕位置センサー
(1 1)	<code>int</code>	<code>waist</code>	腰位置センサー
(1 2)	<code>int</code>	<code>cds</code>	CDS(明るさ)センサー
(1 3)	<code>int</code>	<code>rotaly</code>	ロータリー(足)センサー

第3章 関数リファレンス

3.1 ユーティリティ関数

(1) BN1WaitSec

機能

一定時間ウエイトをかけて処理を一時停止させます。(秒単位)

呼び出し形式

```
void BN1WaitSec(unsigned int time)
```

パラメータ

unsigned int time	設定したい待ち時間 (秒単位) 範囲は 0 秒 ~ 5162 秒
--------------------------	-------------------------------------

ヘッダファイル

bn1api.h

(2) BN1WaitMilliSec

機能

一定時間ウエイトをかけて処理を一時停止させます。(ミリ秒単位)

呼び出し形式

```
void BN1WaitMilliSec(unsigned int time)
```

パラメータ

unsigned int time	設定したい待ち時間 (ミリ秒単位) 範囲は 0 ミリ秒 ~ 65535 ミリ秒
--------------------------	--

ヘッダファイル

bn1api.h

(3) BN1CharToHexStr

機能

1 バイトの char 型の数値を 16 進表記の文字列(2 バイト)にします。

呼び出し形式

```
void BN1CharToHexStr(char *str, unsigned char cval)
```

パラメータ

char	*str	16 進表記に変換した文字列を格納する領域
unsigned char	cval	16 進表記の文字列に変換したい 1 バイトの整数値

ヘッダファイル

bn1api.h

備考

引数 str に渡すポインタの領域は、終端に'¥0'を含めた文字列を格納する為、3 バイト以上の領域を確保する必要があります。

(4) BN1SetTraceLevel

機能

カブリロモニター(capterm)のトレースレベルを設定します。

呼び出し形式

```
void BN1SetTraceLevel(int level)
```

パラメータ

int	level	カブリロモニターで設定するトレースレベル値
-----	-------	-----------------------

ヘッダファイル

bn1api.h

備考

カブリロモニター上で setlevel コマンドによる動的な設定を行うことが出来ますが、プログラムの中で本関数を設定することで強制的にトレースレベルを設定することが可能です。本関数のデフォルトの level は 3 となっています。

(5) BN1TraceOut

機能

BN1TraceOut 関数の第 1 引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、カプリロモニター(capterm)にトレース用文字列を出力します。

呼び出し形式

```
void BN1TraceOut(int level, char *str)
```

パラメータ

int level カプリロモニターで設定するトレースレベル値

char str カプリロモニターで表示するトレース用文字列

ヘッダファイル

bn1api.h

(6) BN1TraceIn

機能

BN1TraceIn 関数の第 1 引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、カブリロモニター(capterm)から文字列を入力することができます。

呼び出し形式

```
void BN1TraceIn(int level, char *str, int strlen)
```

パラメータ

int level	カブリロモニターで設定するトレースレベル値
char *str	カブリロモニターから入力した文字列が入るバッファ
int strlen	str に入る文字列の最大文字数 (* 文字列は最後に'¥0'ヌルが入るので、確保したバッファ-1 のサイズ)

ヘッダファイル

bn1api.h

(7) BN1BreakOut

機能

BN1BreakOut 関数の引数が、BN1SetTraceLevel 関数で設定したレベル値より小さいとき、プログラムを一時中断しカプリアロモニター(capterm)に続行を尋ねます。

呼び出し形式

```
void BN1BreakOut(int level)
```

パラメータ

int level カプリアロモニターで設定するトレースレベル値

ヘッダファイル

bn1api.h

3.2 グラフィックアイ関数

(1) BN1InitEye

機能

グラフィックアイ機能を初期化します。

呼び出し形式

void BN1InitEye(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(2) BN1PutEye

機能

グラフィックアイを出力します。

呼び出し形式

void BN1PutEye(char eyedata)

パラメータ

char eyedata グラフィックアイの出力値

ヘッダファイル

bn1api.h

備考

左右の目の表示方法は本関数を 2 回呼び出すことで実現します。
最初に呼ばれた関数は左目を表示し、2 番目に呼ばれた関数は右目を表示します。

例) BN1PutEye('L'); BN1PutEye('R');

(3) BN1StopEye

機能

グラフィックアイを停止 (消去) します。

呼び出し形式

void BN1StopEye(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(4) BN1SetEyeColor

機能

グラフィックアイの色を設定します。

呼び出し形式

```
void BN1SetEyeColor(int color)
```

パラメータ

int color グラフィックアイの色の設定値

color が	0 のとき	オリジナル
	1 のとき	赤色
	2 のとき	緑色
	3 のとき	橙色

ヘッダファイル

bn1api.h

備考

引数 **color** の値が 0 ~ 3 でセットされたとき、グラフィックアイの色が変更されます。**color** の値が 0 未満、もしくは 3 より大きいときは何も変化しません。

(5) BN1SetEyeBrightness

機能

グラフィックアイの明るさを設定します。

呼び出し形式

```
void BN1SetEyeBrightness(int bright)
```

パラメータ

int bright グラフィックアイの明るさの設定値

bright の値は **0 ~ 7**
 (明 ~ 暗)

ヘッダファイル

bn1api.h

備考

引数 **bright** の値が **0 ~ 7** でセットされたとき、グラフィックアイの明るさが変更されます。**bright** の値が**0**未満、もしくは**7**より大きいときは何も変化しません。

(6) BN1SetEyePattern

機能

グラフィックアイのユーザ定義パターン（外字）を設定します。

呼び出し形式

```
void BN1SetEyePattern(int pnumber, char *pdata)
```

パラメータ

int pnumber グラフィックアイのユーザ定義パターンナンバーの設定値
pnumber の値は 0 ~ 3 の 4 種類

char *pdata 16 個のデータで構成された配列の先頭アドレス

ヘッダファイル

bn1api.h

備考

グラフィックアイは下図のような並びで表示データを構成しています。

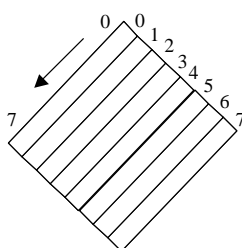
データは 1 行に 1 バイトの 8 列構成ですので、8 バイトで 1 つのパターンを作成しています。

色が赤と緑の 2 つのパターンを同時に作成しますので、ユーザが作成する配列には 16 バイトのデータを用意することになります

グラフィックアイへの送信順序は、

赤 7 赤 6 . . . 赤 0 緑 7 緑 6 . . . 緑 0

であり、赤色と緑色を重ねる合わすことで橙色を表現することも出来ます。



(7) BN1PutEyePattern

機能

BN1SetEyePattern 関数で定義したグラフィックアイのユーザ定義パターン（外字）を出力します。

呼び出し形式

```
void BN1PutEyePattern(int pnumber)
```

パラメータ

int pnumber **BN1SetEyePattern** 関数で定義したユーザ定義パターンナンバーの設定値

pnumber の値は 0 ~ 3 の 4 種類

ヘッダファイル

bn1api.h

備考

左右の目の表示方法は **BN1PutEye** 関数 と同様で、1 番目の関数は左目を表示し、2 番目の関数は右目を表示します。

(8) BN1PlayEyeAnimation

機能

グラフィックアイの定義済みアニメーションを実行します。

呼び出し形式

```
void BN1PlayEyeAnimation(int anumber, int amode)
```

パラメータ

int anumber アニメーション種類の設定値
anumber の値は 1 ~ 15

int amode アニメーションモードの設定値
この設定値でアニメーションが連続か単独か選定出来ます。
amode の値が 1 のとき 連続
amode の値が 1 以外のとき 単独 (1 回)

ヘッダファイル

bn1api.h

備考

アニメーションの種類と設定値の対応は以下の通りです。

設定値：アニメーションの種類

1	：ブロークンハート	(推奨速度 4)
2	：泣く	(推奨速度 3)
3	：燃える	(推奨速度 4)
4	：眠る	(推奨速度 8)
5	：キラーン!	(推奨速度 0)
6	：目が回る	(推奨速度 4)
7	：ラブラブ	(推奨速度 2)
8	：シクシク	(推奨速度 6)
9	：ビックリ!	(推奨速度 1)
10	：目を閉じる	(推奨速度 1)
11	：目を開ける	(推奨速度 1)
12	：通常 ニコニコ	(推奨速度 1)
13	：ニコニコ 通常	(推奨速度 1)
14	：ギョロギョロ	(推奨速度 4)
15	：キョロキョロ	(推奨速度 4)

アニメーションの速度を推奨速度に設定するには後述する BN1SetEyeSpeed 関数の説明を参照下さい。

(9) BN1StopEyeAnimation

機能

グラフィックアイの定義済みアニメーションを停止します。

呼び出し形式

```
void BN1StopEyeAnimation(void)
```

パラメータ

なし

ヘッダファイル

bn1api.h

(10) BN1SetEyeSpeed

機能

グラフィックアイの定義済みアニメーションのスピードを設定します。

呼び出し形式

```
void BN1SetEyeSpeed(int speed)
```

パラメータ

int speed アニメーションのスピード設定値
 speed の値は 0 ~ 15
 (速 ~ 遅)

ヘッダファイル

bn1api.h

備考

本関数は BN1PlayEyeAnimation 関数を呼び出す前でセットして下さい。

3.3 サウンド関数

(1) BN1InitSound

機能

サウンド機能を初期化します。

呼び出し形式

void BN1InitSound(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(2) BN1PlaySound

機能

サウンドの指定番号を演奏します。

呼び出し形式

void BN1PlaySound(int sno)

パラメータ

int sno サウンドの指定番号
(指定番号とサウンドの対応は付録を参照下さい)

ヘッダファイル

bn1api.h

(3) BN1StartSound

機能

サウンドの演奏を開始します。

呼び出し形式

void BN1StartSound(void)

パラメータ

なし

ヘッダファイル

bn1api.h

備考

この関数は主に後述の **BN1StopSound** で停止させたサウンドを再開させたい時に使用します。但し、**BN1StopSound** の停止タイミングによっては再開させた音が乱れる場合があります。必ず正常なサウンドを再演奏させたい場合には、**BN1PlaySound** をお使いください。

(4) BN1StopSound

機能

サウンドの演奏を停止します。

呼び出し形式

void BN1StopSound(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(5) BN1SetSoundVolume

機能

サウンドのボリュームを設定します。

呼び出し形式

```
void BN1SetSoundVolume(int volume)
```

パラメータ

int volume	サウンドボリュームの設定値 volume の値は 0 ~ 31 (音量小 ~ 音量大) デフォルト値は 25
-------------------	--

ヘッダファイル

bn1api.h

(3) BN1DoPosition

機能

指定のポジションにします。

呼び出し形式

```
void BN1DoPosition(int rp, int lp, int wp, int rs, int ls, int ws, int rws, int lws)
```

パラメータ

int rp	右腕位置 (-105 ~ 105) (右腕を右足に寄せた状態 ~ 右腕を全開に上げた状態)
int lp	左腕位置 (-105 ~ 105) (左腕を左足に寄せた状態 ~ 左腕を全開に上げた状態)
int wp	腰位置 (-73 ~ 137) (10 時方向 ~ (反時計回りで) 12 時方向)
int rs	右腕速度 (1 ~ 255) (遅 ~ 速)
int ls	左腕速度 (1 ~ 255) (遅 ~ 速)
int ws	腰速度 (1 ~ 255) (遅 ~ 速)
int rws	右輪速度 (-126 ~ 126) (+数値は前進、-数値は後進) 0 から離れる程速くなる
int lws	左輪速度 (-126 ~ 126) (+数値は前進、-数値は後進) 0 から離れる程速くなる

ヘッダファイル

bn1api.h

(4) BN1DriveRightArm

機能

右腕を単独で動かします。

呼び出し形式

```
void BN1DriveRightArm(int position, int speed)
```

パラメータ

int position 右腕位置 (-105 ~ 105)

int speed 右腕速度 (1 ~ 255)

ヘッダファイル

bn1api.h

(5) BN1DriveLeftArm

機能

左腕を単独で動かします。

呼び出し形式

```
void BN1DriveLeftArm(int position, int speed)
```

パラメータ

int position 左腕位置 (-105 ~ 105)

int speed 左腕速度 (1 ~ 255)

ヘッダファイル

bn1api.h

(6) BN1DriveWaist

機能

腰を単独で動かします。

呼び出し形式

```
void BN1DriveWaist(int position, int speed)
```

パラメータ

int position 腰位置 (-73 ~ 137)

int speed 腰速度 (1 ~ 255)

ヘッダファイル

bn1api.h

(7) BN1DriveRightWheel

機能

右輪を単独で動かします。

呼び出し形式

```
void BN1DriveRightWheel(int speed)
```

パラメータ

int speed 右輪速度 (-126 ~ 126)

ヘッダファイル

bn1api.h

備考

停止させたい場合には本関数の引数 **speed** を 0 にセットして下さい。

(8) BN1DriveLeftWheel

機能

左輪を単独で動かします。

呼び出し形式

```
void BN1DriveLeftWheel(int speed)
```

パラメータ

int speed 右輪速度 (-126 ~ 126)

ヘッダファイル

bn1api.h

備考

停止させたい場合には本関数の引数 speed を 0 にセットして下さい。

(9) BN1StopRightArm

機能

右腕を停止します。(弛緩)

呼び出し形式

```
void BN1StopRightArm(void)
```

パラメータ

なし

ヘッダファイル

bn1api.h

(10) BN1StopLeftArm

機能

左腕を停止します。(弛緩)

呼び出し形式

void BN1StopLeftArm(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(11) BN1StopWaist

機能

腰を停止します。(弛緩)

呼び出し形式

void BN1StopWaist(void)

パラメータ

なし

ヘッダファイル

bn1api.h

3.5 センサー関数

(1) BN1InitSensor

機能

センサー機能を初期化します。

呼び出し形式

void BN1InitSensor(void)

パラメータ

なし

ヘッダファイル

bn1api.h

(2) BN1GetSensorInfo

機能

センサー情報を得ます。

呼び出し形式

```
void BN1GetSensorInfo(SensorInfo_t *info)
```

パラメータ

SensorInfo_t *info センサ情報

ヘッダファイル

bn1api.h

備考 (センサー情報に関する詳細説明)

bn1api.h では、センサー情報を格納する **SensorInfo_t** (typedef をした構造体) を宣言しています。 **SensorInfo_t** のメンバは以下の通りです。

センサー情報一覧

(1)	unsigned char	slant6	傾きセンサー 6 方位
(2)	Bool_t	overturnwarn	転倒警告センサー
(3)	unsigned char	infrared	赤外線センサー
(4)	unsigned char	nikukyu	肉球センサー
(5)	Bool_t	shake	シェイクセンサー
(6)	unsigned char	chira1	チラチラセンサー(前後左右)
(7)	unsigned char	chira2	チラチラセンサー(のど)
(8)	unsigned char	mic	マイクセンサー
(9)	int	leftarm	左腕位置センサー
(1 0)	int	rightarm	右腕位置センサー
(1 1)	int	waist	腰位置センサー
(1 2)	int	cds	CDS(明るさ)センサー
(1 3)	int	rotaly	ロータリー(足)センサー

(3) BN1SensStroke

機能

BN-1 がなでられた状態であるか感知します。

呼び出し形式

Bool_t BN1SensStroke(void)

パラメータ

なし

復帰値

なでられた状態であるならば TRUE が帰ります。それ以外は FALSE が帰ります。

ヘッダファイル

bn1api.h

(4) BN1SensKnock

機能

BN-1 がたたかれた状態であるか感知します。

呼び出し形式

Bool_t BN1SensKnock(void)

パラメータ

なし

復帰値

たたかれた状態であるならば TRUE が帰ります。それ以外は FALSE が帰ります。

ヘッダファイル

bn1api.h

3.6 タイマー割込み関数

(1) BN1SetTimer16

機能

タイマーによる割込み関数を設定します。

(16ビットタイマー 10us 単位で tm count で起動)

呼び出し形式

```
void BN1SetTimer16(void (*func)(void), unsigned long tm)
```

パラメータ

void (*func)(void)	タイマー割込みで呼び出す関数ポインタ
unsigned long tm	タイマーカウントの値

ヘッダファイル

bn1api.h

備考

本関数は第 1 引数に呼び出す関数をセットし、第 2 引数に第 1 引数でセットした関数を呼び出すまでのタイマーカウントを設定します。

例) 16 ビットタイマー 10us 単位でカウントしますので、
第 2 引数に 10000 と設定した場合、 $10\text{us} \times 10000 = 1\text{s}$ となり、
1 秒毎に第 1 引数でセットした関数を呼び出します。

本関数での割込みはセットした関数を一度だけコールします。
続けて割込みを行いたい場合には、ユーザ割込み関数の中で
再び本関数を呼び出してください。

割込みを終了したいときは以下のように記述して下さい。

BN1SetTimer16(NULL, 0)

本割込み機能は BN-1 内部のサーボ制御用に用意された割り込み機能を利用することで実現しております。

よって、サーボ制御にかかる処理時間等の都合により、
精度的に必ずしも正確な割込み時間を保証するものではありません。

(2) BN1SetTimer8

機能

タイマーによる割込み関数を設定します。
(8 ビットタイマー 166ms 単位で tm count で起動)

呼び出し形式

```
void BN1SetTimer8(void (*func)(void), unsigned long tm)
```

パラメータ

void (*func)(void)	タイマー割込みで呼び出す関数ポインタ
unsigned long tm	タイマーカウントの値

ヘッダファイル

bn1api.h

備考

本関数は第 1 引数に呼び出す関数をセットし、第 2 引数に第 1 引数でセットした関数を呼び出すまでのタイマーカウントを設定します。

例) 8 ビットタイマー は 166ms 単位でカウントしますので、
第 2 引数に 10 と設定した場合、 $166\text{ms} \times 10 = 1.66\text{s}$ となり、
1.66 秒毎に第 1 引数でセットした関数を呼び出します。

本関数での割込みはセットした関数を一度だけコールします。
続けて割込みを行いたい場合には、ユーザ関数の中に割込み関数を
呼び出すようなプログラムを組んでください。

割込みを終了したいときは以下のように記述して下さい。

```
BN1SetTimer8(NULL, 0)
```

本関数の割込み時間の精度は、BN1SetTimer16 関数と同様に
必ずしも厳密な精度を保証するものではありません。

3.7 通信関数

(1) BN1WriteCOM

機能

シリアル通信でデータを書き込みます。

呼び出し形式

```
long BN1WriteCOM(int portno, char *buf, long bufsize);
```

パラメータ

int portno	シリアルポート番号
char *buf	送信データの格納バッファ
long bufsize	送信データのサイズ

復帰値

送信したデータのサイズが帰ります。

通常これは引数で指定した `bufsize` と同じ値となります

ヘッダファイル

`bn1api.h`

備考

`portno` の設定値は、BN1-API Version 1.00 では `BN1_PORT_BREAST` 固定です。

また、本関数によるシリアル通信は、送信が完了するまでの間処理をブロックし

送信待ちとなります。

(2) BN1ReadCOM

機能

シリアル通信でデータを読み込みます。

呼び出し形式

```
long BN1ReadCOM(int portno, char *buf, long bufsize)
```

パラメータ

int portno	シリアルポート番号
char *buf	受信データの格納バッファ
long bufsize	受信データのサイズ

復帰値

受信したデータのサイズが帰ります。

通常これは引数で指定した `bufsize` と同じ値となります

ヘッダファイル

`bn1api.h`

備考

`portno` の設定値は、BN1-API Version 1.00 では `BN1_PORT_BREAST` 固定です。

また、本関数によるシリアル通信は、受信が完了するまでの間処理をブロックし受信待ちとなります。

付録

付録A センサー解説

センサーにはレベルセンスセンサーと1ショットセンサーの2種類があります。
レベルセンスセンサーは、読み込み時の現在の状況を常に反映します。
1ショットセンサーは、1度入力があるとデータを読み込むまで保持され、
1度読み込むとリセットされます。

(1) 傾きセンサー 6方位

種類

レベルセンス

受信データと現在の状況

0：計測中

1：正立

2：仰向け

3：顔立ち

4：お尻立ち

5：右横

6：左横

7：エラー

(2) 転倒警告センサー

種類

レベルセンス

受信データと現在の状況

FALSE：正立

TRUE：転倒警報

(3) 赤外線センサー

種類

レベルセンス

反応した赤外線の種類に応じたビットが 1 になるビットフラグです。

受信データと現在の状況

0 bit : 前に障害物がある

1 bit : 後ろに障害物がある

2 bit : 右に障害物がある

3 bit : 左に障害物がある

4 bit : 下に障害物がある

(4) 肉球センサー

種類

レベルセンス

反応した肉球の種類に応じたビットが 1 になるビットフラグです。

受信データと現在の状況

0 bit : 右足に反応あり

1 bit : 左足に反応あり

(5) シェイクセンサー

種類

レベルセンス

受信データと現在の状況

FALSE : 揺されていない

TRUE : 揺されている

(6) チラチラセンサー(前後左右)

種類

レベルセンス

受信データと現在の状況

0: チラチラしない

1: チラチラする

(7) チラチラセンサー(のど)

種類

レベルセンス

受信データと現在の状況

0: チラチラしない

1: チラチラする

(8) マイクセンサー

種類

1ショット

受信データと現在の状況

0: マイク入力なし

1: 1回入力あり

2: 2回入力あり

3: 常時入力あり(うるさい)

(9) 左腕位置センサー

種類

レベルセンス

受信データと現在の状況

0~3FF: 左腕を左足に寄せた状態~左腕を全開に上げた状態

(1 0) 右腕位置センサー

種類

レベルセンス

受信データと現在の状況

0 ~ 3FF : 右腕を右足に寄せた状態 ~ 右腕を全開に上げた状態

(1 1) 腰位置センサー

種類

レベルセンス

受信データと現在の状況

0 ~ 3FF : 10 時方向 ~ (反時計回りで) 12 時方向

(1 2) CDS(明るさ)センサー

種類

レベルセンス

受信データと現在の状況

0 ~ 3FF : 暗 ~ 明

(1 3) ロータリー(足)センサー

種類

レベルセンス

受信データと現在の状況

0 ~ 3FF : 回転した位置情報

付録B データ解説

(1) モーションデータ

BN1DoMotion 関数の引数で指定する番号とモーション名の対応は以下の通りです。

表2 . モーションデータ一覧表 (その1)

No.	モーション名称	No.	モーション名称
1	お座り(姿勢6)	36	中速右後カーブ
2	起立(姿勢8)	37	中速左後カーブ
3	伏せ(姿勢1)	38	高速右回転
4	回転姿勢(姿勢9)	39	高速左回転
5	中腰(姿勢5)	40	高速右カーブ
6	急ブレーキ(姿勢4)	41	高速左カーブ
7	のぞく(姿勢11)	42	高速右後カーブ
8	両手上げ45°	43	高速左後カーブ
9	両手上げ90°	44	歩回転右(早)
10	空白(お座り)	45	歩回転左(早)
11	ゆっくりお座り	46	歩回転右(普)
12	空白(お座り)	47	歩回転左(普)
13	ゆっくり伏せ	48	歩回転右(遅)
14	右足あげ	49	歩回転左(遅)
15	左足あげ	50	歩回転右(大)
16	伏せ右手あげ	51	歩回転左(大)
17	伏せ左手あげ	52	歩回転右(小)
18	姿勢1(伏せ)	53	歩回転左(小)
19	姿勢2	54	歩行右カーブR小
20	姿勢3	55	歩行左カーブR小
21	姿勢4(急ブレーキ)	56	歩行右カーブR大
22	姿勢5(中腰)	57	歩行左カーブR大
23	姿勢6(お座り)	58	歩行右後カーブR小
24	姿勢7	59	歩行左後カーブR小
25	姿勢8(起立)	60	歩行右後カーブR大
26	姿勢9(回転姿勢)	61	歩行左後カーブR大
27	姿勢10	62	左足上げ歩行
28	姿勢11(のぞく)	63	右足上げ歩行
29	ゆっくり両手上げ45°	64	中速前進
30	空白(お座り)	65	高速前進
31	空白(お座り)	66	歩行前普通
32	中速右回転	67	歩行前早い
33	中速左回転	68	歩行前ゆっくり
34	中速右カーブ	69	歩行前チョコ2小股
35	中速左カーブ	70	歩行前ゆったり大股

表3 . モーションデータ一覧表 (その2)

No.	モーション名称	No.	モーション名称
71	歩行前チョコ2大股	119	歩行後チョコ2大股
72	走行前進1	120	走行後退1
73	走行前進2	121	走行後退2
74	歩行前ゆったり小股	122	歩行後ゆったり小股
75	空白(お座り)	123	空白(お座り)
76	空白(お座り)	124	空白(お座り)
77	空白(お座り)	125	フリフリ走行後退
78	スケーティング1	126	エビバック
79	スケーティング2	127	押し逃げ
80	スキップ歩行	128	右向く
81	空白(お座り)	129	左向く
82	うさぎ歩行	130	右向け右
83	うさぎ走行	131	パレード
84	ほぶく前進	132	パレード前進
85	引き寄せ	133	パレード停止
86	空白(お座り)	134	パレード後退
87	チョコチョコ走り	135	パレード手おろす
88	フリフリ走行	136	右肩を落とす
89	ハイハイ歩き	137	左肩を落とす
90	空白(お座り)	138	上体回体操(速)
91	行進歩き	139	上体回体操(遅)
92	やあやあ歩き	140	上体回体操(極遅)
93	わいわい歩き	141	空白(お座り)
94	空白(お座り)	142	空白(お座り)
95	空白(お座り)	143	空白(お座り)
96	空白(お座り)	144	お手
97	空白(お座り)	145	おかわり
98	空白(お座り)	146	チンチン
99	空白(お座り)	147	ハイ!
100	空白(お座り)	148	尻立ち
101	空白(お座り)	149	仰向けになる
102	空白(お座り)	150	ワーイ
103	空白(お座り)	151	ねえねえ
104	空白(お座り)	152	ゴメン
105	空白(お座り)	153	バイバイ1
106	空白(お座り)	154	バイバイ2
107	空白(お座り)	155	ヤダヤダ1
108	空白(お座り)	156	ヤダヤダ2
109	空白(お座り)	157	ダダ1
110	空白(お座り)	158	ダダ2
111	空白(お座り)	159	右キック
112	中速後退	160	左キック
113	高速後退	161	降参
114	歩行後普通	162	ノリノリ
115	歩行後早い	163	悔しい
116	歩行後ゆっくり	164	いじけ虫
117	歩行後チョコ2小股	165	失神
118	歩行後ゆったり大股	166	ソワソワ

表4 . モーションデータ一覧表 (その3)

No.	モーション名称	No.	モーション名称
167	猫パンチ	211	横ノリ
168	猫パンチワンツー	212	伸び
169	排出中	213	仰向けで伸び
170	排出ブルブル	214	空白(お座り)
171	バーン!	215	空白(お座り)
172	でんぐり	216	フリフリ走行(遅)
173	がえし	217	千鳥足歩行
174	でんぐりがえし	218	ツルツル足踏み
175	バーン! 2	219	空白(お座り)
176	ブルブル	220	空白(お座り)
177	ありがとう	221	寝姿1伏手後
178	右手あげて	222	寝姿2仰向手上げ
179	左手あげて	223	寝姿3上体捻り寝
180	腕立て一回	224	寝姿4伏手後肩落し
181	ごろにゃん	225	寝姿1・4用寝返
182	モジモジ	226	寝姿2用寝返
183	起き上がり1	227	寝姿3用寝返
184	後手から前手	228	空白(お座り)
185	寝起き	229	空白(お座り)
186	バンザイ	230	伏寝姿左足上寝返
187	右手おろす	231	伏寝姿右足上寝返
188	左手おろす	232	空白(お座り)
189	フラダンス	233	空白(お座り)
190	倒れ起き	234	空白(お座り)
191	もがく	235	ハットスピン肩中心右回転
192	右手でやあ	236	ハットスピン額中心右回転
193	左手でやあ	237	空白(お座り)
194	逆立ち	238	空白(お座り)
195	オヨヨ歩行	239	空白(お座り)
196	三三七拍子	240	空白(お座り)
197	三三七拍子(直立)	241	空白(お座り)
198	寝起き2	242	空白(お座り)
199	倒れ起き2	243	空白(お座り)
200	民族ダンス	244	空白(お座り)
201	3歩歩いて2歩下がる	245	空白(お座り)
202	あくび	246	空白(お座り)
203	お尻ダンス	247	空白(お座り)
204	オヨヨダンス	248	空白(お座り)
205	空白(お座り)	249	空白(お座り)
206	ハットスピン肩中心左回転	250	空白(お座り)
207	ハットスピン額中心左回転	251	空白(お座り)
208	ヤダヤダ3	252	空白(お座り)
209	ヤダヤダ4	253	空白(お座り)
210	ヤッター!	254	空白(お座り)

(2) サウンドデータ

BN1PlaySound 関数の引数で指定する番号とモーション名の対応は以下の通りです。

表5 . サウンドデータ一覧表

No.	サウンド名称	No.	サウンド名称
0	嬉しい(小)	33	ファンファーレ(中)
1	嬉しい(中)	34	ファンファーレ(大)
2	嬉しい(大)	35	ゲームオーバー
3	嬉しい(長)	36	悲しい(普2)
4	悲しい(小)	37	怒り(普)
5	悲しい(中)	38	驚き(普)
6	悲しい(大)	39	ニヤー(元気)
7	悲しい(長)	40	ニヤー(恐怖)
8	怒り(小)	41	ニヤー(淋)
9	怒り(中)	42	ニヤー(あまえ)
10	怒り(大)	43	ニヤー(いじけ)
11	怒り(長)	44	ニヤー(喜び)
12	怖い(小)	45	ニヤー(怒)
13	怖い(中)	46	ニヤー(悲)
14	怖い(大)	47	ニヤー(不思議)
15	怖い(長)	48	ピッ
16	驚き(小)	49	ポッ
17	驚き(中)	50	パッ
18	驚き(大)	51	ブー
19	驚き(長)	52	キーン
20	不思議(小)	53	ピ
21	不思議(中)	54	ブ
22	不思議(大)	55	ピーー
23	不思議(長)	56	ピー
24	あまえる(小)	57	猫ふんじゃった
25	あまえる(中)	58	アルプス一万尺
26	あまえる(大)	59	三三七拍子
27	あまえる(長)	60	Air
28	嬉しい(普1)	61	ワルキューレ騎行
29	嬉しい(普2)	62	葬送行進曲
30	嬉しい(普3)	63	ノクターンop9-2
31	悲しい(普1)	64	カルメン
32	ファンファーレ(小)	65	運命

(3) グラフィックアイ(アニメーション)データ

BN1PlayEyeAnimation関数で指定する第1引数の番号とアニメーションとの対応は以下の通りです。

表6. アニメーションデータ一覧表

No.	アニメーション名称	推奨速度
1	ブローケンハート	4
2	泣く	3
3	燃える	4
4	眠る	8
5	キラーン!	0
6	目が回る	4
7	ラブラブ	2
8	シクシク	6
9	ビックリ!	1
10	目を閉じる	1
11	目を開ける	1
12	通常 ニコニコ	1
13	ニコニコ 通常	1
14	ギョロギョロ	4
15	キョロキョロ	4